

MACKMA Architektur

Die Architektur des MACKMA-Systems ist modular aufgebaut, so dass Erweiterungen und Anpassungen flexibel vorgenommen werden können. Im Folgenden werden die einzelnen Bestandteile der Architektur genauer vorgestellt. Zur Visualisierung werden UML-Modelle verwendet.

MACKMA Core

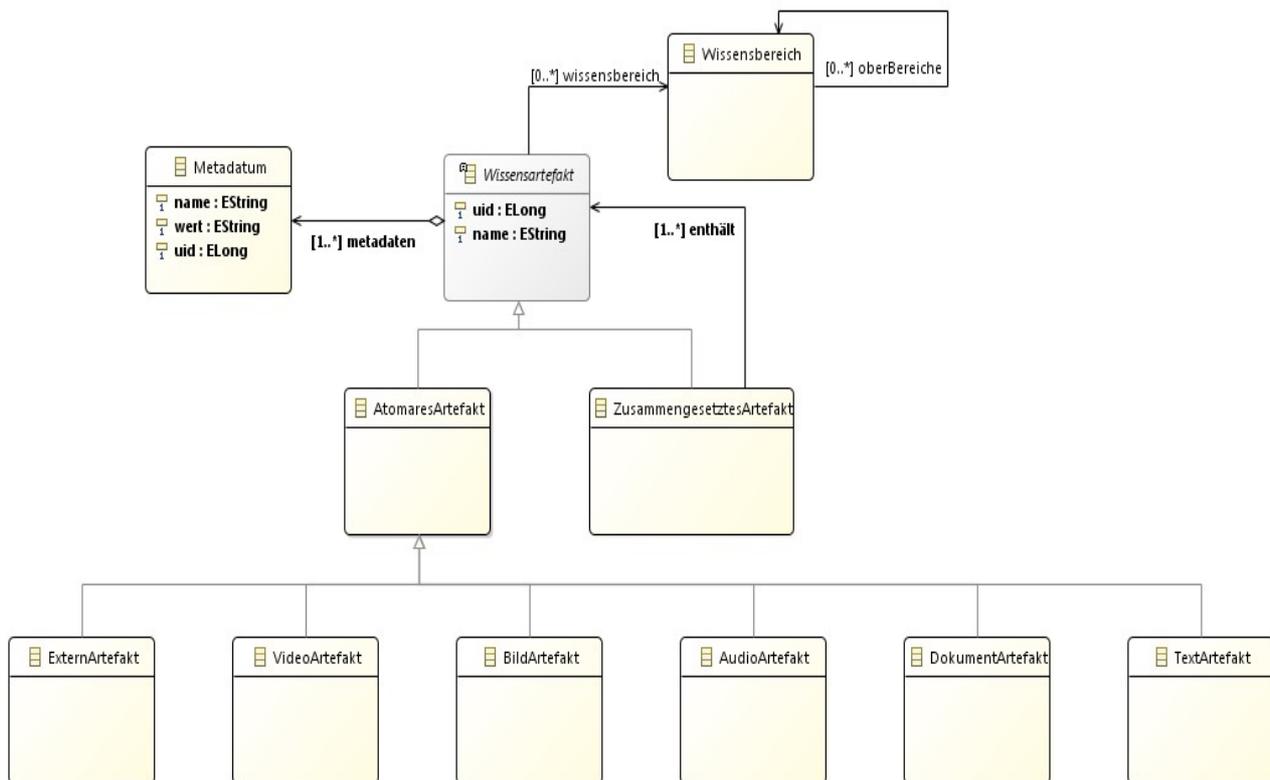


Abbildung 1: Architektur MACKMA Core

Atomare Artefakte

Das Kernmodul der MACKMA-Architektur ist in Abbildung 1 dargestellt. Im Zentrum steht die Klasse *Wissensartefakt*. Diese ist als abstrakte Basisklasse für alle möglichen Arten von Wissensartefakten zu sehen. Ein einzelnes Artefakt wird mittels der Klasse *AtomaresArtefakt* dargestellt. Je nach Typ wird ein Wissensartefakt mit der entsprechenden Klasse instanziiert:

- Die Klasse *TextArtefakt* repräsentiert einzelne Textbausteine, die genutzt werden, um Wissen zu beschreiben. Der Inhalt dieser Artefakte lässt sich anhand gängiger Formatierungen (Absätze, Überschriften etc.) strukturieren.
- Mittels der Klasse *DokumentArtefakt* wird ein Textdokument mit einem bestimmten Dateitypen (PDF, OOP etc.) repräsentiert. Dazu wird der Inhalt des Dokuments vollständig in das WMS integriert, d.h. für den Nutzer ist nicht transparent, dass es sich um eine Datei im Dateisystem handelt.
- Die Klassen *BildArtefakt*, *AudioArtefakt* und *VideoArtefakt* repräsentieren Dateien des entsprechenden Typs. Sie werden vollständig in das WMS integriert, so dass z.B. Videos in ein

Wissensartefakt eingebettet werden können. Durch die vollständige Integration in das WMS profitieren diese Arten von Wissensartefakten von der Versionierung und ermöglichen Benachrichtigungen bei Änderungen. So werden bei Änderungen an den Dateien, welche den Wissensartefakten zugrunde liegen, automatisch die entsprechenden Wissensartefakte aktualisiert.

- Die Klasse *ExternArtefakt* ermöglicht die Referenzierung externer Dateien, die nicht im WMS integriert sind. Dadurch können Nutzer die entsprechenden Datei z.B. herunterladen und mit einem Programm ihrer Wahl bearbeiten. Hierunter fallen auch Zugriffe auf andere Systeme mittels API-Aufrufen, z.B. die Anzeige von Tickets eines redmine-Systems. Im Unterschied zu Dokument-, Bild-, Audio- und Videoartefakten unterstehen externe Artefakte nicht der Kontrolle des WMS. Dementsprechend können Nutzer z.B. keine Benachrichtigungen erhalten, wenn die Artefakte geändert werden.

Zusammengesetzte Artefakte

Mit Hilfe der aufgezählten spezifischen Artefakttypen lassen sich einzelne Wissensinhalte in das WMS übertragen. Um verschiedene Artefakte miteinander zu kombinieren, wird die Klasse *ZusammengesetztesArtefakt* verwendet. Mit Hilfe dieser Klasse, können Wissensartefakte ineinander verschachtelt werden, so dass sich auch komplexere Inhalte darstellen lassen. Hierbei ist es möglich, Wissensartefakte verschiedener Typen zu kombinieren. Z.B. kann ein Wissensartefakt Text mit einem Video kombiniert werden, um verschiedene Beschreibungsebenen bereitzustellen. Anhand der Strukturierung in Abbildung 1 ist ersichtlich, dass sich auch zusammengesetzte Artefakte wieder in neuen Artefakten verwenden lassen.

Wissensbereiche

Um eine weitere Strukturierung der Wissensartefakte zu ermöglichen, können diese mittels der Klasse *Wissensbereich* spezifischen Wissensbereichen zugeordnet werden. Ein Wissensbereich kann verschiedene Strukturierungsmerkmale eines Unternehmens betreffen, z.B. kann ein Wissensbereich Unternehmensbereiche adressieren während ein anderer Wissensbereich das angebotene Portfolio strukturiert. Um dies zu ermöglichen, lassen sich Wissensbereiche hierarchisch anordnen.

Ein Wissensartefakt lässt sich einer beliebigen Anzahl von Wissensbereichen zuordnen. Dadurch ist sichergestellt, dass die verschiedenen Strukturierungsansätze mit den entsprechenden Artefakten verbunden sind, z.B. lässt sich ein Artefakt sowohl dem Wissensbereich *Marketing* (Strukturierung anhand Unternehmensbereichen als auch dem Wissensbereich *Druckinformationen* (Strukturierung anhand des Portfolios) zuordnen.

Metadaten

Metadaten lassen sich verwenden, um Wissensartefakte genauer zu beschreiben. Um größtmögliche Flexibilität zu ermöglichen, repräsentiert die Klasse *Metadaten* zunächst als einfache Schlüssel-Wert-Paare. Dies ermöglicht es, Metadaten generischer Datenformate in Textform zu erstellen (Text, Datum, Referenzen etc.). Die effektivere Nutzung von Metadaten wird ermöglicht, indem ein unternehmensweit vereinheitlichter Katalog von Metadaten bereitgestellt wird. Dieser enthält vorgegebene Namen und Beschreibungen einzelner Metadaten, die genutzt werden dürfen.

MACKMA Rollen

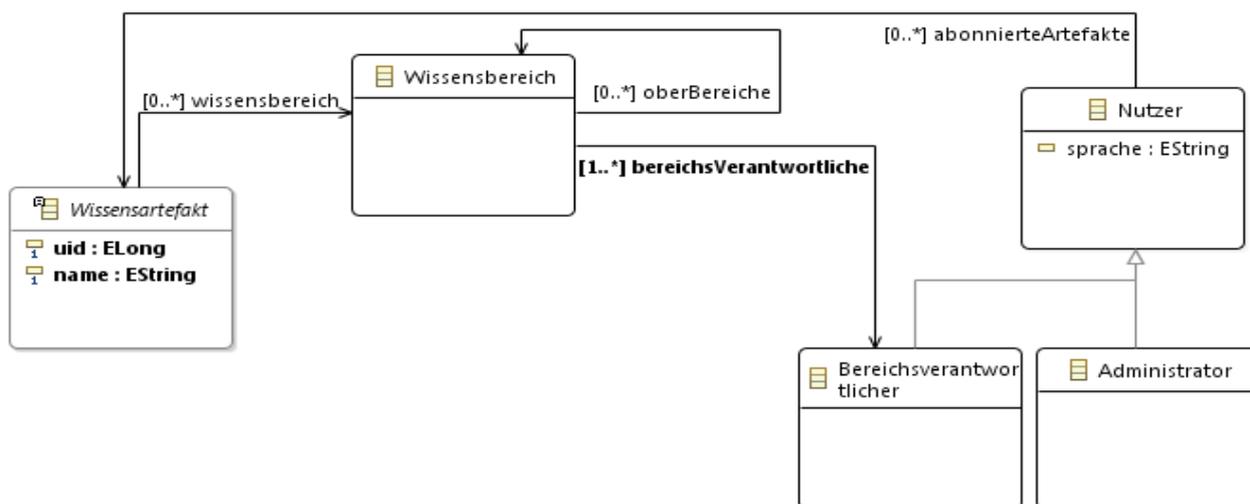


Abbildung 2: Architektur MACKMA Rollen

Nutzer

Für das Rollenkonzept ist die Abbildung 2 dargestellte Klasse *Nutzer* zentral. Diese repräsentiert eine Person, die das WMS nutzt und auf Wissensartefakte sowohl lesend als auch schreibend zugreifen kann. Jeder Nutzer kann eine beliebige Anzahl von Wissensartefakten *abonnieren* und wird damit über Änderungen am entsprechenden Artefakt benachrichtigt. Um eine Mehrsprachigkeit des Systems zu ermöglichen, haben alle Nutzer ein Attribut *sprache*. Dieses legt fest, in welcher Sprache Wissensartefakte bevorzugt präsentiert werden. Existieren verschiedene Übersetzungen eines Artefakts wird dementsprechend die jeweils passende angezeigt.

Um die technische und inhaltliche Verwaltung des WMS zu trennen, werden zwei besondere Nutzer spezifiziert. Hier existiert zunächst die Klasse *Bereichsverantwortlicher*, unter die Nutzer fallen, die für die inhaltliche Richtigkeit eines Wissensbereich verantwortlich zeichnen. Darüber hinaus sind mit der Klasse *Administrator* jene Nutzer gekennzeichnet, die für die technische Wartung des WMS verantwortlich zeichnen.

Bereichsverantwortliche

Jedem Wissensbereich können ein oder mehrere Bereichsverantwortliche zugeordnet werden. Diese stellen sicher, dass alle Wissensartefakte des entsprechenden Bereichs aktuell sind. Darüber hinaus können sie Änderungen an Wissensartefakten nach einer Begutachtung freigeben, so dass sich die Wissensbasis stets in einem konsistenten Zustand befindet. Bereichsverantwortliche sind Ansprechpartner für Fragen von Systemnutzern, z.B. wenn diese auf Ungenauigkeiten oder Fehler beim Lesen eines Wissensartefakts stoßen.

MACKMA Prozess

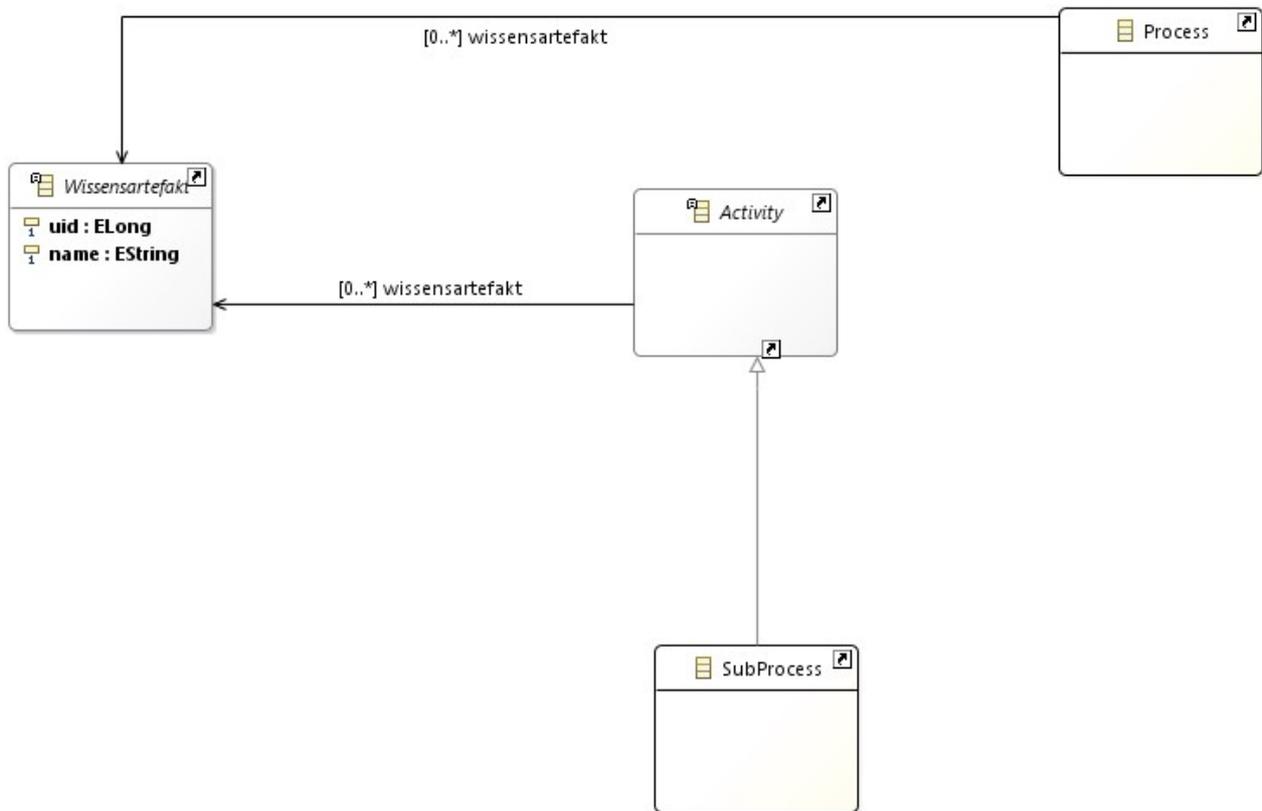


Abbildung 3: Architektur MACKMA Prozess

Aktivitäten und Prozesse

Mit Hilfe der Klassen aus dem Prozessmodul lassen sich Wissensartefakte mit Geschäftsprozessen verknüpfen. Entsprechend der Darstellung in Abbildung 3 ist zu erkennen, dass hier die Elemente der Notation BPMN verwendet werden. Dazu werden die Klassen *Activity*, *Process* und *Subprocess* genutzt. Sowohl Aktivitäten als auch Prozessen können eine beliebige Anzahl von Wissensartefakten zugewiesen werden. Diese werden genutzt, um die Aufgaben, welche durch die Prozesselemente beschrieben werden, genauer zu spezifizieren. Durch die Verschachtelung der Prozesselemente (z.B. Aktivitäten A, B und C gehören zu Prozess P) ergibt sich automatisch die Verschachtelung der entsprechenden Wissensartefakte.

Eine beispielhaft Zuweisung von Wissensartefakten zu Prozessbestandteilen ist in Abbildung 4 dargestellt. Dort existiert der Prozess *A*, welcher die Aktivitäten *A1*, *A2* und *A3* enthält. Die Aktivität *A3* ist wiederum ein Subprozess, welcher die Aktivitäten *A11* und *A12* enthält. Im Beispiel sind die beiden Wissensartefakte *WA1* und *WA2* dem Prozess *A* zugewiesen. Der Aktivität *A12* ist hingegen das zusammengesetzte Wissensartefakt *ZWA34* zugewiesen.

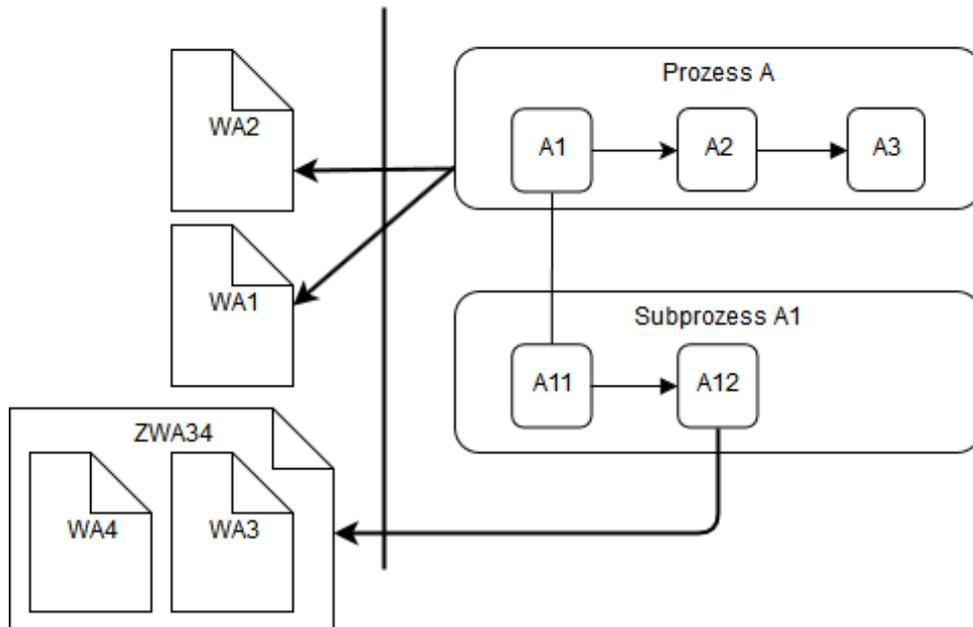


Abbildung 4: Zuweisung Wissensartefakte zu Prozessbestandteilen

Erweiterungen

Aufbauend auf dem existierenden Metamodell lassen sich verschiedene Erweiterungen der Funktionalität implementieren. Dadurch können z.B. wichtige Metadaten statt über die angegebene Klasse direkt in das Metamodell integriert werden. Dazu sind die entsprechenden Klassen, Attribute und Relationen zu ergänzen. Beispiele für mögliche Erweiterungen sind:

- Internationalisierung: Die Klasse *Nutzer* enthält bereits das Attribut *sprache*, welches für die weitere Verwendung in der Internationalisierung genutzt werden kann, Dazu ist das Metadatum *Language* in die Attributliste eines Wissensartefakts zu übernehmen. Nun kann durch einen Abgleich der beiden Sprachen festgestellt werden, welches Wissensartefakt angezeigt werden soll. Daneben kann die Relation *übersetzungVon* genutzt werden, um zwei Wissensartefakte miteinander zu verknüpfen. Damit ist sichergestellt, dass bei Änderungen eines der beiden Artefakte das andere immer mit aktualisiert werden muss.
- Semantisches Tagging: Durch Erweiterung der Klasse *Wissensartefakt* um das Attribut *tags* können einzelne Artefakte genauer beschrieben bzw. klassifiziert werden. Im Gegensatz zur Nutzung von Tags über Metadaten als einfaches Name-Wert-Tupel ermöglicht die Formalisierung als Attribut effizientere Validierungen.

Metadaten

Metadatum	Kategorie	Automatisierbar	Beschreibung
Erstellungsdatum	Lifecycle	ja	Erstellungsdatum des Wissensartefakts.
Änderungsdatum	Lifecycle	ja	Datum der letzten Änderung
Abonnent	Lifecycle	nein	Durch Nutzer durchgeführtes Abonnieren von WA, um bei Änderungen an denen benachrichtigt zu werden.
Contributor	Lifecycle	ja	Nutzer, welcher etwas zu dem WA beigetragen hat.
Creator	Lifecycle	ja	Ersteller des WAs
Verwendungsform	Lifecycle	ja	Was wurde mit dem WA gemacht (geändert, gelöscht,...)
Wissensbereich	Structure	nein	Zuordnung des WA zu einem oder mehreren Wissensbereichen
Prozess	Structure	Nein	Zuordnung des WA zu einem oder mehreren Prozessen
Links	Structure	teilweise	Relationen zwischen WA definieren. Mögliche Relationen könnten sein: (wird) referenziert/(wird) benötigt/ergänzt...
Tags	Structure	nein	Nutzervergebene Schlagworte, welche den Inhalt beschreiben
Versionsnummer	Structure	ja	Zahl, welche eine spezifische Version eines WA referenziert
Identifizier	Structure	ja	Zahl/Zeichenkette/URI, welche ein Wissensartefaktcontainer eindeutig bezeichnet Wissensartefaktcontainer: Sammlung eines Wissensartefakts mit allen seinen Versionen
Zugriffsrechte	Security/Quality	nein	Vergabe von Rechten zum Lesen, Änderung, Löschen und ggf. Freigeben/Verstecken von WA
Freigabe	Security/Quality	nein	Änderungen müssen explizit von Personen/Gruppe freigegeben werden
Hidden	Security/Quality	nein	Wenn WA "hidden" ist, ist es nicht gelöscht, aber auch nicht sichtbar
Error	Security/Quality	nein	WA wurde durch Nutzer als fehlerhaft gekennzeichnet
Rating	Content	nein	Nutzen/Qualität durch User definiert
Language	Content	nein	Sprache des WA
Comment	Content	nein	Kommentar zum WA
Description	Content	teilweise	Beschreibung des WA
Format	Content	ja	Format einer Datei
Title	Content	nein	Titel eines WAs
Pflichtzugriff	Recipient	nein	Sicherstellung des Wissenstransfers durch Lesepflicht für bestimmte User für bestimmte WA